164

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 454 495 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
07.02.1996 Bulletin 1996/06

(51) Int Cl.[6]: H04N 1/40

(21) Application number: 91303820.4

(22) Date of filing: 26.04.1991

(54) Half-tone image processing system

Halbton-Bildverarbeitungssystem

Système de traitement d'images en demi-teintes

(84) Designated Contracting States:
DE FR GB

(30) Priority: 26.04.1990 JP 111320/90
21.05.1990 JP 130701/90
25.05.1990 JP 133847/90
25.05.1990 JP 133851/90

(43) Date of publication of application:
30.10.1991 Bulletin 1991/44

(73) Proprietor: FUJITSU LIMITED
Kawasaki-shi, Kanagawa 211 (JP)

(72) Inventor: Itoh, Seiichi
Kawasaki-shi, Kanagawa 214 (JP)

(74) Representative:
Fane, Christopher Robin King et al
London, WC2A 1AT (GB)

(56) References cited:
- PATENT ABSTRACTS OF JAPAN, vol. 14, no. 26 (M-921)[3969], 18th January 1990; & JP-A-1 264 846
- PATENT ABSTRACTS OF JAPAN, vol. 9, no. 214 (P-384)[1937], 31st August 1985; & JP-A-60 74 091
- PATENT ABSTRACTS OF JAPAN, vol. 12, no. 417 (P-782)[3264], 7th November 1988; & JP-A-63 155 273
- PATENT ABSTRACTS OF JAPAN, vol. 12, no. 494 (P-805)[3341], 23rd December 1988; & JP-A-63 204 375
- PATENT ABSTRACTS OF JAPAN, vol. 12, no. 111 (E-598)[2958], 8th April 1988; & JP-A-62 239 666

EP 0 454 495 B1

Description

The present invention relates generally to image data processing and in particular to digital image processing apparatus for use, for example, in digital copy machines, image scanners, facsimile machines and the like.

In general, an image is constituted by a plurality of pixels each of which is formed by density data consisting of a plurality of bits. The pseudo half-tone image is basically obtained by binary-coding the pixel in accordance with a predetermined method.

As is known, for example, a photograph includes many half-tone images. However, it is very difficult and troublesome to reproduce the half-tone image from the original photograph. Two methods have been proposed for reproducing a half-tone image, namely an "ordered dither" method and an error diffusion method.

There are, however, some problems in these methods as explained in detail hereinafter.

It is desirable to provide a half-tone image processing system enabling high quality half-tone image reproduction of an original image.

According to a first aspect of the present invention, there is provided image data processing circuitry for processing image data items, representative of the respective density values of pixels making up an original image, to derive output data items for determining the density values of pixels making up a corresponding binary output image, which circuitry includes:

storage means for storing successive sets of such image data items relating respectively to such original image pixels, each of which sets is made up of a data item relating to an object pixel and further data items relating respectively to peripheral pixels, adjacent to the said object pixel;
calculation means connected to the said storage means and operable to receive image data items of such a stored set and to produce a modified object pixel data item derived from the said image data item relating to the said object pixel of the set in dependence upon the respective differences in density values between that object pixel image data item and such further data items of the set concerned;
comparison means connected to receive the modified object pixel data item and operable to compare the density value of that data item with a preset threshold value and, in dependence upon the result of such comparison, to determine such an output data item corresponding to the object pixel concerned; and
error correction means operable to determine an object pixel error representative of the difference in density values between the said modified object pixel data item and the determined output data item and operable also, in dependence upon that error, to modify such further data items in the said storage means so as to enable the circuitry to compensate for such a determined object pixel error when processing a subsequent set of such image data items.

According to a second aspect of the present invention, there is provided image data processing circuitry for processing image data items, representative of the respective density values of pixels making up an original image, to derive output data items for determining the density values of pixels making up a corresponding binary output image, which circuitry includes:

storage means for storing successive sets of such image data items relating respectively to such original image pixels, each of which sets is made up of a data item relating to an object pixel and further data items relating respectively to peripheral pixels, adjacent to the said object pixel;
calculation means connected to the said storage means and operable to receive image data items of such a stored set and to produce a difference data item in dependence upon the respective differences in density values between the object pixel data item of the set and such further data items of the set concerned;
comparison means connected to receive the said image data item relating to the said object pixel and the said difference data item and operable to compare the respective density values of those data items and, in dependence upon the result of such comparison, to determine such an output data item corresponding to the object pixel concerned; and
error correction means operable to determine an object pixel error representative of the difference in density values between the said image data item relating to the said object pixel and the determined output data item and operable also, in dependence upon that error, to modify such further data items in the said storage means so as to enable the circuitry to compensate for such a determined object pixel error when processing a subsequent set of such image data items.

According to a third aspect of the present invention, there is provided image data processing circuitry for processing image data items, representative of the respective density values of pixels making up an original image, to derive output data items for determining the density values of pixels making up a corresponding binary output image, which circuitry includes:

storage means for storing successive sets of such image data items relating respectively to such original image pixels, each of which sets is made up of a data item relating to an object pixel and further data items relating respectively to peripheral pixels, adjacent to the said object pixel;
calculation means connected to the said storage means and operable to receive image data items of such a stored set and to produce a modified object pixel data item derived from the said image data item relating to the said object pixel of the set in dependence upon the respective differences in density values between that object pixel image data item and such further data items of the set concerned and also in dependence upon a correction value for the object pixel of the set concerned;
comparison means connected to receive the modified object pixel data item and operable to compare the density value of that data item with a preset threshold value and, in dependence upon the result of such comparison, to determine such an output data item corresponding to the object pixel concerned; and
error correction means operable to determine an object pixel error representative of the difference in density values between the said modified object pixel data item and the determined output data item and operable also to store that error for use by the circuitry in calculating the correction value for the object pixel of a subsequent set of such image data items, thereby to enable the circuitry to compensate for such a determined object pixel error when processing that subsequent set.

According to a fourth aspect of the present invention, there is provided image data processing circuitry for processing image data items, representative of the respective density values of pixels making up an original image, to derive output data items for determining the density values of pixels making up a corresponding binary output image, which circuitry includes:

storage means for storing successive sets of such image data items relating respectively to such original image pixels, each of which sets is made up of a data item relating to an object pixel and further data items relating respectively to peripheral pixels, adjacent to the said object pixel;
calculation means connected to the said storage means and operable to receive image data items of such a stored set and to produce a difference data item in dependence upon the respective differences in density values between the object pixel image data item of the set and such further data items of the set concerned and also operable to produce a corrected object pixel data item derived from the said object pixel data item in dependence upon a correction value for that object pixel;
comparison means connected to receive the said corrected object pixel data item and the said difference data item and operable to compare the respective density values of those data items and, in dependence upon the result of such comparison, to determine such an output data item corresponding to the object pixel concerned; and
error correction means operable to determine an object pixel error representative of the difference in density values between the said corrected object pixel data item and the determined output data item and operable also to store that error for use by the circuitry in calculating the correction value for the object pixel of a subsequent set of such image data items, thereby to enable the circuitry to compensate for such a determined object pixel error when processing that subsequent set.

Reference will now be made, by way of example, to the accompanying drawings, in which:

Fig. 1 is a view for explaining an ordered dither method;
Fig. 2A to 2c are views for explaining an error diffusion method;
Fig. 3 is a schematic block diagram of a half-tone image processing system according to a first embodiment of the present invention;
Fig. 4 is a block diagram of the half-tone image processing system shown in Fig. 3;
Fig. 5 is a partially detailed block diagram of the half-tone image processing system of Fig. 3;
Fig. 6 is a detailed block diagram of the Laplacian calculation unit of Fig. 3;
Fig. 7 is a detailed block diagram of the error distribution unit of Fig. 3;
Figs. 8A to 8C are examples of Laplacian coefficient matrixes;
Fig. 9 is a schematic block diagram of half-tone image processing system according to a second embodiment of the present invention;
Fig. 10 is a schematic block diagram of half-tone image processing system according to a third embodiment of the present invention; and
Fig. 11 is a schematic block diagram of half-tone image processing system according to a fourth embodiment of the present invention.

Figure 1 is a view for explaining an ordered dither method. In Fig. 1, the left side denotes an original image, the

center denotes a dither matrix having a threshold, and the right side denotes a binary-coded image.

In the original image, each square denotes one pixel, and a numeral in each square denotes the density of the pixel.

In the dither matrix, the matrix enclosed by a thick solid line denotes a 4 × 4 dither matrix. A numeral in each square denotes the threshold value of the density of each pixel. The 4 × 4 matrix is used as a general size of the dither matrix. As is obvious, all 4 × 4 dither matrixes have the same order of numerals as the matrix enclosed by the thick line. Although they are not shown, there are mainly three types of 4 × 4 dither matrices, namely a Bayer type, a meshed-point type and a spiral type. The difference between these types lies only in the order of numerals defining the threshold value.

As is obvious from the drawing, the binary-coded image is constituted by a plurality of black and white dots, and is obtained in such a manner that a black dot is defined when the density data of the pixel is equal to or larger than the threshold value, and a white dot is defined when the density data of the pixel is smaller than the threshold value.

There are, however, some problems in the ordered dither method as explained below.

First, some striped patterns (so-called "moire" pattern) periodically occur in the image. This striped pattern deteriorates the quality of the reproduced image, particularly, when the original image is a printed image consisting of the meshed-point. In general, the striped pattern may be caused by the periodicity of sequences (threshold values) of dither matrixes. This is because all 4 × 4 dither matrixes have the same sequence of numerals.

Second, when the original image contains characters and line drawings, parts of lines of the characters and line drawings are periodically cut-off so that the reproduced image becomes worse.

Third, the larger the matrix size is, the worse the resolution of the image is. That is, to obtain multi-gradation density, when the matrix size of the half-tone image processing system is set to larger size than general matrix size, it is difficult to achieve both the multi-gradation and high resolution since the resolution of the reproduced image becomes lower than the resolution of a scanner.

Figures 2A to 2c are views for explaining an error diffusion method. The error diffusion method is well-known as a method enabling multi-gradation and high resolution. This method is disclosed, for example, in the publication "An Adaptive Algorithm for Spatial Grey Scale" by R.W. Floyd and L. Steinberg, 1975 SID International Symposium Digest of Technical Papers. 4.3. pp 36-37, April 1975.

In Fig. 2A, one square denotes one pixel, and D11 to D33 denote density data each having eight bits. The center pixel (slant line portion) having the density D22 is an object pixel. In Fig. 2B, since the density data is expressed by eight bits, 0 to 255 denote the density gradation of the pixel. The value 127 is a center value of the density gradation. Assuming that the density of the object pixel D22 has the value 225, the difference between the top value 255 and the value 225 is given as an error value -30.

In Fig. 2C, the binary-coding operation is performed by weighting the density data as explained below. That is, blank squares corresponding to the pixels D11 to D21, which are positioned before the center pixel, are the pixels for which the binary-coding operation has been completed. The pixels D23, D31, D32 and D33 positioned after the center pixel D22 have not yet been subjected to the binary-coding operation. Each of these pixels is weighted by corresponding numerals 2, 1, 2 and 1. In this case, the weighted value 2 is assigned to the nearest pixels D23 and D32, and the weighted value 1 is assigned to the other pixels D31 and D33. The error value -30 is assigned to the peripheral pixels as follows. The value 6 is a sum of all weighted values 2, 1, 2 and 1.

D23: $-30 \times 2/6 = -10$
D31: $-30 \times 1/6 = -5$
D32: $-30 \times 2/6 = -10$
D33: $-30 \times 1/6 = -5$

Accordingly, the new density values of the pixels D23 to D33 are assigned as follows.

D23' = D23 -10
D31' = D31 - 5
D32' = D32 -10
D33' = D33 - 5

The new density values D23', D31', D32' and D33' are binary-coded by the threshold value, for example, the center density value 127.

As is obvious from the above explanation, the error diffusion method is more advantageous than the ordered dither method in that the striped pattern (moire pattern) does not occur in the reproduced image, and the multi-gradation and the resolution are improved.

There are, however, some problems in this error diffusion method as explained below.

First, in comparison with a simple binary-coded image, the quality of the reproduced image is not sufficiently close to the original image in the reproduction of characters or line drawings.

Second, in the low density gradation, since a dot is changed to a bit "1" after error values are gradually accumulated, a great deal of time is necessary for reproduction of the image dots.

Accordingly, it is desirable to provide a half-tone image processing system enabling a high quality pseudo half-tone image.

Figure 3 is a schematic block diagram of a half-tone image processing system according to a first embodiment of the present invention. In Fig. 3, reference number 1 denotes a density data memory, 2 a Laplacian calculation unit, 3 a binary-coding unit, 4 an error calculation unit, 5 an error distribution unit, and 6 a binary data output unit. Further, TH_1 denotes a threshold value for binary-coding. In the first embodiment, the threshold value TH_1 is preset as a fixed value.

In Fig. 3, the density data memory 1 stores the density data of each dot of the original image. In this case, the density data of each dot is constituted by a plurality of bits, for example, eight bits. The Laplacian calculation unit 2 performs a Laplacian calculation between the density data of the object pixel and the density data of the peripheral pixels around the object pixel to obtain corrected data for the object pixel. The Laplacian calculation is performed in such a manner that a difference between the density of the object pixel and the density of each peripheral pixel is obtained and all of the differences are summed. After the Laplacian calculation, the density data of the peripheral pixels are corrected by the error diffusion matrix for weighting and diffusing the original density data as explained in detail below. for the object pixel, produced by

The corrected data DA for the object pixel, produced by the Laplacian calculation, is applied to the binary-coding unit 3 and the error calculation unit 4, and compared with the threshold value TH_1. In the binary-coding unit 3, when the corrected object pixel data DA is equal to or larger than the threshold value TH_1, the binary-coding unit 3 outputs the bit "1". When the corrected data DA is smaller than the threshold value TH_1, binary-coding unit 3 outputs the bit "0".

The error calculation unit 4 calculates the binary-coded error ER between the corrected data DA and the output of the binary-coding unit 3. In this case, when the output of the binary-coding unit 3 is the bit "1", the maximum value of the density data is input to the error calculation unit 4. When the output of the binary-coding unit 3 is the bit "0", the minimum value of the density data is input to the error calculation unit 4.

The error ER is output from the error calculation unit 4 to the error distribution unit 5. The error distribution unit 5 distributes the binary-coded error ER to density data of the peripheral pixel (the pixel to be processed) based on the error diffusion coefficient, and adds the distributed error value to the corresponding original pixel to revise the density data of the original pixel.

In this embodiment, since there is no boundary portion of the gradation in comparison with the ordered dither method, the profile of a character or line drawing is emphasized so that it is possible to achieve a high quality reproduced image.

Figure 4 is a block diagram of the half-tone image processing system shown in Fig. 3, Figure 5 is a partially detailed block diagram of the half-tone image processing system shown in Fig. 3, Figure 6 is a detailed block diagram of the Laplacian calculation unit shown in Fig. 3, Figure 7 is a detailed block diagram of the error distribution unit shown in Fig. 3, and Figures 8A to 8C are examples of Laplacian coefficient matrixes used in the Laplacian calculation of the binary-coded error.

In Fig. 4, reference number 11 denotes a density data line buffer (RAM) corresponding to the density data memory 1 in Fig. 3. Reference number 14 denotes an adder corresponding to the error calculation unit 4 in Fig. 3. Reference number 17 also denotes an adder included in the error distribution unit 5 in Fig. 3. Reference number 18 denotes an error diffusion matrix.

The original image is read out by a line-type image sensor (not shown) having N (integer) sensing elements for every line, and is digitized in accordance with the density data. As previously explained, the density data of each pixel is constituted by eight bits so that it is possible to express the density data of the pixel by 256 (integer) density gradations (from 0 to 255). That is, the minimum density is the integer 0, and the maximum density is the integer 255.

In the density data line buffer 11, $D_{m-1,n-1}$, $D_{m-1,n}$, --- $D_{m+1,n+1}$ represent the density data of each of the pixels. These pixels correspond to the matrix shown in Fig. 8A, and the data $D_{m,n}$ represents the density data of the object pixel. The density data of the object and the peripheral pixels are input to the Laplacian calculation unit 12. The following calculation is performed in the Laplacian calculation unit 12.

$$D_{m,n}' = D_{m,n} + P((D_{m,n} - D_{m-1,n}) + (D_{m,n} - D_{m,n-1}) + (D_{m,n} - D_{m,n+1}) + (D_{m,n} - D_{m+1,n})) \qquad (1)$$

Here, P is a positive value. As is obvious from the above formula, the Laplacian calculation is performed for the difference between the object pixel (m,n) and the peripheral pixels (m-1,n; m,n-1; m,n+1; m+1,n) around the object pixel.

Based on the corrected object pixel data ($D_{m,n}'$) produced by the above calculation, the density data of the object and peripheral pixels are corrected as explained below. The corrected data (modified object pixel data item) $D_{m,n}'$ is input to the binary-coding unit 13, and is compared with the preset threshold value TH_1.

In the binary-coding unit 13, when the corrected data $D_{m,n}'$ is larger than the threshold value TH_1, the binary-coded density data ($O_{m,n}$) of the object pixel (m,n) is expressed by the value 255 (black). When the corrected data $D_{m,n}'$ is equal to or smaller than the threshold value TH_1, the binary-coded density data ($O_{m,n}$) of the object pixel $D_{m,n}$ is ex-

pressed by the value 0 (white). The binary-coded data $O_{m,n}$ is than output from the binary-data output unit 16. That is, when the data $O_{m,n}$ is "255", the output value is the bit "1" (black), and when the data $O_{m,n}$ is "O", the output value is the bit "O" (white).

The binary-coded data $O_{m,n}$ is also output from the binary-coded unit 13 to the adder 14. In the adder 14, the binary-coded error $E_{m,n}$ is obtained as the difference between the corrected data $D_{m,n}'$ and the binary-coded data $O_{m,n}$ as shown by the following formula.

$$E_{m,n} = D_{m,n}' - O_{m,n} \tag{2}$$

This formula is a feature of the first embodiment. Further, the binary-coded error $E_{m,n}$ is input to the error distribution unit 15, is weighted by the error diffusion coefficient K, and is distributed to the peripheral pixels (m,n+1; m+1,n-1; m+1,n; m+1,n+1) in accordance with the error diffusion matrix 18. Accordingly, the density data of the peripheral pixels (m,n+1: m+1,n-1; m+1,n; m+1,n+1) are corrected as follows.

$$D_{m,n+1}' = D_{m,n+1} + (K_{m,n+1} / \Sigma K_{i,j}) \times E_{m,n}$$

$$D_{m+1,n-1}' = D_{m+1,n-1} + (K_{m+1,n-1} / \Sigma K_{i,j}) \times E_{m,n}$$

$$D_{m+1,n}' = D_{m+1,n} + (K_{m+1,n} / \Sigma K_{i,j}) \times E_{m,n}$$

$$D_{m+1,n+1}' = D_{m+1,n+1} + (K_{m+1,n+1} / \Sigma K_{i,j}) \times E_{m,n}$$

Where,

$$\Sigma K_{i,j} = K_{m,n+1} + K_{m+1,n-1} + K_{m+1,n} + K_{m+1,n+1}$$

As is obvious from the above, in the Figure 4 system the binary-coded error $E_{m,n}$ is used for obtaining the corrected density data of the peripheral pixels.

Figure 5 is a partially detailed block diagram of the half-tone image processing system of Fig. 3, Figure 6 is a detailed block diagram of the Laplacian calculation unit of Fig. 3, and Figure 7 is a detailed block diagram of the error distribution unit of Fig. 3.

In Fig. 5, reference numbers 31 and 32 denote density data line buffers corresponding to the density data line buffer 11 in Fig. 4, each density data line buffer is constituted by a RAM functioning as a FIFO (first-in first-out) buffer. Reference numbers 33 to 40 denote latch circuits for storing the data, 41 to 44 denote adders, 45 denotes a Laplacian calculation unit corresponding to the Laplacian calculation unit 12 in Fig. 4, 46 denotes an error distribution unit corresponding to the error distribution unit 15 in Fig. 4, 47 denotes binary-coding circuit, and 48 denotes an adder for error calculation. In Fig. 6, reference numbers 50 to 56, 58 denote adders, and reference number 57 denotes a conversion table. Further, in Fig. 7, reference numbers 59 to 62 denote conversion tables (for example, ROM).

The operation of the system is explained in detail below. In Fig. 5, density data $D_{1,n}$ of the pixel is input to the latch circuit 33, and sequentially transferred through the latch circuits 34 to 40, the adders 41 to 44 and the density data line buffers 31 and 32. Initially, in all latch circuits, the Q output is set to "O", and the inverted Q($\bar{Q}$) output is set to "1".

The density data $D_{m,n}$ of the latch circuit 37, i.e., the density data of the object pixel (m,n) is input to the Laplacian calculation unit 45. Further, the density data of the peripheral pixels, i.e., the density data $D_{m-1,n}$ of the latch circuit 40, the density data $D_{m,n-1}$ of the latch circuit 38, the density data $D_{m,n+1}$ of the latch circuit 36, and the density data $D_{m+1,n}$ of the latch circuit 34 are also input to the Laplacian calculation unit 45.

The density data $D_{m,n}$ is added to each density data $D_{m-1,n}$, $D_{m,n-1}$, $D_{m,n+1}$ and $D_{m+1,n}$ by using each of the adders 50 to 56 as shown in Fig. 6. The resultant data of the adder 56 is input to the conversion table 57. The conversion table 57 previously stores the resultant data from multiplying the Laplacian coefficient by the density data. The adder 58 adds the density data $D_{m,n}$ of the object pixel (m,n) to the output of the conversion table 57 so that the adder 58 outputs the corrected density data $D_{m,n}'$.

The corrected density data $D_{m,n}'$ is input to the binary-coding circuit 47 as shown in Fig. 5, and is compared with the threshold value TH_1. The binary-coding circuit 47 outputs the binary data as follows.

When the corrected density data $D_{m,n}'$ is larger than the threshold value TH_1 in the inverted $\bar{O}$ output ($\bar{O}_{m,n}$), all eight bits are "O", and in the O output ($O_{m,n}$), the single bit is "1".

That is, the binary-coded output ($O_{m,n}$) is "1", and the density data becomes maximum (all eight bits are "1") so that all eight bits become "O" as the complement of the output "1".

When the corrected density data $D_{m,n}'$ is equal to or smaller than the threshold value TH_1, in the inverted O output ($\bar{O}_{m,n}$), all eight bits are "1", and in the O output ($O_{m,n}$), the single bit is "O".

That is, the binary-coded output ($O_{m,n}$) is "O", and the density data becomes maximum (all eight bits are "O") so that all eight bits become "1" as the complement of the output "O".

The binary-coding circuit 47 is constituted by, for example, a comparator and a NOT circuit (both not shown). The binary-coded error $E_{m,n}$ is obtained by the adder 48 is based on the density data $D_{m,n}$ and the inverted output $O_{m,n}$. The binary-coded error $E_{m,n}$ is input to the error distribution unit 46 as shown in Fig. 7. Each of the conversion tables 59 to 62 stores the error distribution value which is previously obtained from the weighted coefficient of the error diffusion matrix 18 shown in Fig. 4, and outputs the error distribution values ($E_{m,n+1}$, $E_{m+1,n-1}$, $E_{m+1,n}$, $E_{m+1,n+1}$). These error distribution values are input to the adders 41 to 44 as shown in Fig. 5.

As shown in Fig. 5, the adders 41 to 44 add the Q output of the latch circuits 33 to 36 ($D_{m,n+1}$, $D_{m+1,n-1}$, $D_{m+1,n}$, $D_{m+1,n+1}$) to the error distribution values ($E_{m,n+1}$, $E_{m+1,n-1}$, $E_{m+1,n}$, $E_{m+1,n+1}$), respectively.

Figures 8A to 8C are examples of Laplacian coefficient matrixes used in the Laplacian calculation of the binary-coded error. In Figs. 8A to 8C, each number denotes a Laplacian coefficient for weighting the density data as previously explained. In these drawings, the center pixel denotes an object pixel to be weighted. In case of the Laplacian matrix shown in Fig. 8C, many peripheral pixels are provided around the object pixel (+36), but the Laplacian calculation is the same as shown in Fig. 8A.

Figure 9 is a schematic block diagram of a half-tone image processing system according to a second embodiment of the present invention. In Fig. 9, the same reference numbers as used in Fig. 4 are attached to the same components in this drawing. As is obvious from the drawing, the structure of this embodiment is basically the same structure as that of the first embodiment shown in Fig. 4. In this embodiment, the density data $D_{m,n}$ of the object pixel (m,n) is input to the binary-coding unit 13 and the adder 14. Further, the resultant data D_x (difference data item) of the Laplacian calculation is used as the threshold value TH_2. In this case, the resultant data D_X is obtained from the formula (1). That is, the data D_x is given by $P((D_{m,n} - D_{m-1,n}) + \cdots + (D_{m,n} - D_{m+1,n}))$, i.e. $TH_2 = D_x = P((D_{m,n} - D_{m-1,n}) + (D_{m,n} - D_{m,n-1}) + (D_{m,n} - D_{m,n+1}) + D_{m,n} - D_{m+1,n}))$. Accordingly, the threshold value TH_2 is variable in this embodiment.

In the binary-coding unit 13, the density data $D_{m,n}$ of the object pixel (m,n) is compared with the threshold value TH_2. That is, when the density data $D_{m,n}$ is larger than the threshold value TH_2, the binary-coded density data $O_{m,n}$ is expressed by the value 255 (black). When the density data $D_{m,n}$ is equal to or smaller than the threshold value TH_2, the binary-coded density data $O_{m,n}$ is expressed by the value O (white). The binary-coded density data $O_{m,n}$ is output to the binary-data output unit 16. That is, when the data $O_{m,n}$ is "255", the output value from the binary-data output unit 16 is "1" (black), and when the data $O_{m,n}$ is "O", the output value is "O" (white).

The binary-coded data $O_{m,n}$ is also output from the binary-coding unit 13 to the adder 14. In the adder 14, the binary-coded error $E_{m,n}$ is obtained as the difference between the density data $D_{m,n}$ and the binary-coded data $O_{m,n}$ as shown by the following formula.

$$E_{m,n} = D_{m,n} - O_{m,n} \tag{3}$$

The binary-coded error $E_{m,n}$ is input to the error distribution unit 15. The subsequent explanations are omitted since, after this stage, the calculations are the same as those of the first embodiment shown in Fig. 4.

Figure 10 is a schematic block diagram of a half-tone image processing system according to a third embodiment of the present invention. In Fig. 10, the same reference numbers as used in Fig. 4 are attached to the same components in this drawing. Reference number 19 denotes an error weighted matrix, 20 an error data line memory, 21 an average error calculation unit, and 22 an adder.

As shown in the drawing, the density data $D_{m,n}$ of the object pixel (m,n) is input to the adder 22, and the resultant data D_x of the Laplacian calculation is also input to the adder 22. In this case, the resultant data D_x is obtained from the formula (1). That is, the data D_x is given by $P((D_{m,n} - D_{m-1,n}) + \cdots + (D_{m,n} - D_{m+1,n}))$. Further, a weighted average value (i.e. correction value) $C_{m,n}$ is input from the average error calculation unit 21 to the adder 22. The correction value $C_{m,n}$ is obtained by the following calculation in the average error calculation unit 21. That is,

$$C_{m,n} = \frac{((K_{m-1,n-1} \times E_{m-1,n-1}) + (K_{m-1,n} \times E_{m-1,n}) + (K_{m-1,n+1} \times E_{m-1,n+1}) + (K_{m,n-1} \times E_{m,n-1}))}{(K_{m-1,n-1} + K_{m-1,n} + K_{m-1,n+1} + K_{m,n-1})} \tag{4}$$

The density data $D_{m,n}$, the resultant data D_x, and the correction value $C_{m,n}$ are added to each other in the adder 22, and the adder 22 outputs the resultant data $D_{m,n}$" to the binary-coding unit 13 and the adder 14. In the binary-coding unit 13, the resultant data $D_{m,n}$" is compared with a threshold value TH_3. The threshold value TH_3 is preset as a fixed value.

When the resultant data $D_{m,n}$" is larger than the threshold value TH_3, the binary-coded density data $O_{m,n}$ is expressed by the value 255 (black). When the resultant data $D_{m,n}$" is equal to or smaller than the threshold value TH_3, the binary-coded density data $O_{m,n}$ is expressed by the value O (white). The binary-coded data $O_{m,n}$ is output to the binary-data output unit 16. That is, when the data $O_{m,n}$ is "255", the output value from the binary-data output unit 16 is "1" (black), and when the data $O_{m,n}$ is "O", the output value is "O" (white).

The binary-coded data $O_{m,n}$ is also input from the binary-coding unit 13 to the adder 14. In the adder 14, the binary-coded error $E_{m,n}$ is obtained as the difference between the resultant data $D_{m,n}$" and the binary-coded data $O_{m,n}$ as

shown by the following formula.

$$E_{m,n} = D_{m,n}'' - O_{m,n} \quad (5)$$

Accordingly, the binary-coded error $E_{m,n}$ of the formula (5) is input to the error data line buffer 20 so that the correction value $C_{m,n}$ can be obtained from the formula (4) in the average error calculation unit 21 as explained above.

Figure 11 is a schematic block diagram of a half-tone image processing system according to a fourth embodiment of the present invention. In fig. 11, the same reference numbers as used in Fig. 10 are attached to the same components in this drawing. As is obvious from the drawing, the structure of this embodiment is basically the same structure as that of the third embodiment shown in Fig. 10. In this embodiment, the density data $D_{m,n}$ of the object pixel (m,n) is input to the adder 22. Further, the resultant data (difference data item) D_x of the Laplacian an calculation is used as the threshold value TH_4. Accordingly, threshold value TH_4 is variable in this embodiment.

As is obvious from the drawing, the density data $D_{m,n}$ of the object pixel (m,n) and the correction value $C_{m,n}$ are input to the adder 22, and the adder 22 outputs the resultant data $D_{m,n}'''$ to the binary-coding unit 13. Since the binary-coded data $O_{m,n}$ from the binary-coding unit 13 is obtained in the same manner as explained in Fig. 10, the explanation is omitted in this embodiment. The binary-coded data $O_{m,n}$ is also input from the binary-coding unit 13 to the adder 14. In the adder 14, the binary-coded error $E_{m,n}$ is obtained as the difference between the resultant data $D_{m,n}'''$ and the binary-coded data $O_{m,n}$ as shown by the following formula.

$$E_{m,n} = D_{m,n}''' - O_{m,n} \quad (6)$$

Accordingly, the binary-coded error $E_{m,n}$ of the formula (6) is input to the error data line buffer 20 so that the correction value $C_{m,n}$ can be obtained from the formula (4) in the average error calculation unit 21 as explained above.

Claims

1. Image data processing circuitry for processing image data items, representative of the respective density values of pixels making up an original image, to derive output data items for determining the density values of pixels making up a corresponding binary output image, which circuitry includes:

   storage means (11) for storing successive sets of such image data items relating respectively to such original image pixels, each of which sets is made up of a data item ($D_{m,n}$) relating to an object pixel and further data items ($D_{m-1,n-1}$, ..., $D_{m+1,n+1}$) relating respectively to peripheral pixels, adjacent to the said object pixel;
   calculation means (12) connected to the said storage means (11) and operable to receive image data items of such a stored set and to produce a modified object pixel data item ($D_{m,n}'$) derived from the said image data item ($D_{m,n}$) relating to the said object pixel of the set in dependence upon the respective differences in density values between that object pixel image data item ($D_{m,n}$) and such further data items ($D_{m-1,n}$, $D_{m,n-1}$, $D_{m,n+1}$, $D_{m+1,n}$) of the set concerned;
   comparison means (13) connected to receive the modified object pixel data item ($D_{m,n}'$) and operable to compare the density value of that data item ($D_{m,n}'$) with a preset threshold value (TH_1) and, in dependence upon the result of such comparison, to determine such an output data item ($O_{m,n}$) corresponding to the object pixel concerned; and
   error correction means (14, 15) operable to determine an object pixel error ($E_{m,n}$) representative of the difference in density values between the said modified object pixel data item ($D_{m,n}'$) and the determined output data item ($O_{m,n}$) and operable also, in dependence upon that error ($E_{m,n}$), to modify such further data items in the said storage means (11) so as to enable the circuitry to compensate for such a determined object pixel error when processing a subsequent set of such image data items.

2. Image data processing circuitry for processing image data items, representative of the respective density values of pixels making up an original image, to derive output data items for determining the density values of pixels making up a corresponding binary output image, which circuitry includes:

   storage means (11) for storing successive sets of such image data items relating respectively to such original image pixels, each of which sets is made up of a data item ($D_{m,n}$) relating to an object pixel and further data items ($D_{m-1,n-1}$, ..., $D_{m+1,n+1}$) relating respectively to peripheral pixels, adjacent to the said object pixel;
   calculation means (12) connected to the said storage means (11) and operable to receive image data items of such a stored set and to produce a difference data item (D_x) in dependence upon the respective differences in density values between the object pixel data item ($D_{m,n}$) of the set and such further data items ($D_{m-1,n}$, $D_{m,n-1}$,

$D_{m,n+1}$, $D_{m+1,n}$) of the set concerned;
comparison means (13) connected to receive the said image data item ($D_{m,n}$) relating to the said object pixel and the said difference data item (D_x) and operable to compare the respective density values of those data items ($D_{m,n}$, D_x) and, in dependence upon the result of such comparison, to determine such an output data item ($O_{m,n}$) corresponding to the object pixel concerned; and
error correction means (14, 15) operable to determine an object pixel error ($E_{m,n}$) representative of the difference in density values between the said image data item ($D_{m,n}$) relating to the said object pixel and the determined output data item ($O_{m,n}$) and operable also, in dependence upon that error ($E_{m,n}$), to modify such further data items in the said storage means (11) so as to enable the circuitry to compensate for such a determined object pixel error when processing a subsequent set of such image data items.

3. Image data processing circuitry as claimed in claim 1 or 2, wherein the said error correction means (14, 15) include an error distribution unit (15) connected for receiving the determined object pixel error ($E_{m,n}$) and also connected for receiving a predetermined error diffusion matrix (K) and operable, for each peripheral pixel of the said set that follows the said object pixel, to calculate a weighted error value (for example $K_{m-1,n-1} \times E_{m,n}$) based on the determined object pixel error ($E_{m,n}$) and the weight value (for example $K_{m-1,n-1}$) in the said error diffusion matrix (K) which corresponds to that peripheral pixel;
the circuitry further including data item modifying means (17) connected to the said error correction means (14, 15) for receiving therefrom the respective weighted error values for the peripheral pixels following the object pixel and operable to modify the further data items relating to those pixels in dependence respectively upon the weighted error values.

4. Image data processing circuitry for processing image data items, representative of the respective density values of pixels making up an original image, to derive output data items for determining the density values of pixels making up a corresponding binary output image, which circuitry includes:

storage means (11) for storing successive sets of such image data items relating respectively to such original image pixels, each of which sets is made up of a data item ($D_{m,n}$) relating to an object pixel and further data items $D_{m-1,n-1}$, ..., $D_{m+1,n+1}$) relating respectively to peripheral pixels, adjacent to the said object pixel;
calculation means (12, 22) connected to the said storage means (11) and operable to receive image data items of such a stored set and to produce a modified object pixel data item ($D_{m,n}''$) derived from the said image data item ($D_{m,n}$) relating to the said object pixel of the set in dependence upon the respective differences in density values between that object pixel image data item ($D_{m,n}$) and such further data items ($D_{m-1,n}$, $D_{m,n-1}$, $D_{m,n+1}$, $D_{m+1,n}$) of the set concerned and also in dependence upon a correction value ($C_{m,n}$) for the object pixel of the set concerned;
comparison means (13) connected to receive the modified object pixel data item ($D_{m,n}''$) and operable to compare the density value of that data item ($D_{m,n}''$) with a preset threshold value (TH_3) and, in dependence upon the result of such comparison, to determine such an output data item ($O_{m,n}$) corresponding to the object pixel concerned; and
error correction means (14, 19-21) operable to determine an object pixel error ($E_{m,n}$) representative of the difference in density values between the said modified object pixel data item ($D_{m,n}''$) and the determined output data item ($O_{m,n}$) and operable also to store that error for use by the circuitry in calculating the correction value for the object pixel of a subsequent set of such image data items, thereby to enable the circuitry to compensate for such a determined object pixel error when processing that subsequent set.

5. Image data processing circuitry for processing image data items, representative of the respective density values of pixels making up an original image, to derive output data items for determining the density values of pixels making up a corresponding binary output image, which circuitry includes:

storage means (11) for storing successive sets of such image data items relating respectively to such original image pixels, each of which sets is made up of a data item ($D_{m,n}$) relating to an object pixel and further data items ($D_{m-1,n-1}$, ..., $D_{m+1,n+1}$) relating respectively to peripheral pixels, adjacent to the said object pixel;
calculation means (12, 22) connected to the said storage means (11) and operable to receive image data items of such a stored set and to produce a difference data item (D_x) in dependence upon the respective differences in density values between the object pixel image data item ($D_{m,n}$) of the set and such further data items of the set concerned and also operable to produce a corrected object pixel data item ($D_{m,n}'''$) derived from the said object pixel data item ($D_{m,n}$) in dependence upon a correction value ($C_{m,n}$) for that object pixel;
comparison means (13) connected to receive the said corrected object pixel data item ($D_{m,n}'''$) and the said

difference data item (D_x) and operable to compare the respective density values of those data items ($D_{m,n}'''$, D_x) and, in dependence upon the result of such comparison, to determine such an output data item ($O_{m,n}$) corresponding to the object pixel concerned; and
error correction means (14, 15) operable to determine an object pixel error ($E_{m,n}$) representative of the difference in density values between the said corrected object pixel data item ($D_{m,n}'''$) and the determined output data item ($O_{m,n}$) and operable also to store that error for use by the circuitry in calculating the correction value for the object pixel of a subsequent set of such image data items, thereby to enable the circuitry to compensate for such a determined object pixel error when processing that subsequent set.

6. Image data processing circuitry as claimed in claim 4 or 5, wherein the said error correction means (14, 19-21) are operable, for each peripheral pixel of the said set that precedes the said object pixel, to retrieve the object pixel error ($E_{m-1,n-1}$, $E_{m-1,n}$, $E_{m-1,n+1}$, $E_{m,n-1}$) stored by the error correction means when processing a previous set of the image data items that had as its said object pixel the peripheral pixel concerned;

the said error correction means including an average error calculation unit connected for receiving a predetermined error weighted matrix (K) and operable to combine the retrieved errors in dependence upon respective corresponding weight values in the said error weighted matrix (K) to calculate the said correction value ($C_{m,n}$).

Patentansprüche

1. Bilddatenverarbeitungsschaltungsanordnung zum Verarbeiten von Bilddatenelementen, die für die jeweiligen Dichtewerte von Pixels, die ein Originalbild bilden, repräsentativ sind, um Ausgabedatenelemente zum Bestimmen der Dichtewerte von Pixels abzuleiten, die ein entsprechendes binäres Ausgabebild bilden, welche Schaltungsanordnung enthält:

ein Speichermittel (11) zum Speichern von sukzessiven Sätzen solcher Bilddatenelemente, die sich jeweilig auf solche Originalbildpixels beziehen, von welchen Sätzen jeder gebildet ist aus einem Datenelement ($D_{m,n}$), das sich auf ein Objektpixel bezieht, und weiteren Datenelementen ($D_{m-1,n-1}$, ..., $D_{m+1,n+1}$), die sich jeweilig auf periphere Pixels beziehen, die an das genannte Objektpixel angrenzen;
ein Berechnungsmittel (12), das mit dem genannten Speichermittel (11) verbunden ist und betriebsfähig ist, um Bilddatenelemente solch eines gespeicherten Satzes zu empfangen und ein abgewandeltes Objektpixeldatenelement ($D_{m,n}'$) zu erzeugen, das von dem genannten Bilddatenelement ($D_{m,n}$) abgeleitet ist, das sich auf das genannte Objektpixel des Satzes bezieht, in Abhängigkeit von den jeweiligen Differenzen von Dichtewerten zwischen jenem Objektpixelbilddatenelement ($D_{m,n}$) und solch weiteren Datenelementen ($D_{m-1,n}$, $D_{m,n-1}$, $D_{m,n+1}$, $D_{m+1,n}$) des betreffenden Satzes;
ein Vergleichsmittel (13), das verbunden ist, um das abgewandelte Objektpixeldatenelement ($D_{m,n}'$) zu empfangen, und betriebsfähig ist, um den Dichtewert jenes Datenelementes ($D_{m,n}'$) mit einem voreingestellten Schwellenwert (TH_1) zu vergleichen und um in Abhängigkeit von dem Resultat solch eines Vergleichs solch ein Ausgabedatenelement ($O_{m,n}$) zu bestimmen, das dem betreffenden Objektpixel entspricht; und
ein Fehlerkorrekturmittel (14, 15), das betriebsfähig ist, um einen Objektpixelfehler ($E_{m,n}$) zu bestimmen, der für die Differenz von Dichtewerten zwischen dem genannten abgewandelten Objektpixeldatenelement ($D_{m,n}'$) und dem bestimmten Ausgabedatenelement ($O_{m,n}$) repräsentativ ist, und auch betriebsfähig ist, um in Abhängigkeit von jenem Fehler ($E_{m,n}$) solche weiteren Datenelemente in dem genannten Speichermittel (11) abzuwandeln, so daß die Schaltungsanordnung solch einen bestimmten Objektpixelfehler kompensieren kann, wenn ein nachfolgender Satz solcher Bilddatenelemente verarbeitet wird.

2. Bilddatenverarbeitungsschaltungsanordnung zum Verarbeiten von Bilddatenelementen, die für die jeweiligen Dichtewerte von Pixels, die ein Originalbild bilden, repräsentativ sind, um Ausgabedatenelemente zum Bestimmen der Dichtewerte von Pixels abzuleiten, die ein entsprechendes binäres Ausgabebild bilden, welche Schaltungsanordnung enthält:

ein Speichermittel (11) zum Speichern von sukzessiven Sätzen solcher Bilddatenelemente, die sich jeweilig auf solche Originalbildpixels beziehen, von welchen Sätzen jeder gebildet ist aus einem Datenelement ($D_{m,n}$), das sich auf ein Objektpixel bezieht, und weiteren Datenelementen ($D_{m-1,n-1}$, ... $D_{m+1,n+1}$), die sich jeweilig auf periphere Pixels beziehen, die an das genannte Objektpixel angrenzen;
ein Berechnungsmittel (12), das mit dem genannten Speichermittel (11) verbunden ist und betriebsfähig ist, um Bilddatenelemente solch eines gespeicherten Satzes zu empfangen und um ein Differenzdatenelement (D_x)

in Abhängigkeit von den jeweiligen Differenzen von Dichtewerten zwischen dem Objektpixeldatenelement ($D_{m,n}$) des Satzes und solch weiteren Datenelementen ($D_{m-1,n}$, $D_{m,n-1}$, $D_{m,n+1}$, $D_{m+1,n}$) des betreffenden Satzes zu erzeugen;

ein Vergleichsmittel (13), das verbunden ist, um das genannte Bilddatenelement ($D_{m,n}$), das sich auf das genannte Objektpixel bezieht, und das genannte Differenzdatenelement (D_x) zu empfangen, und betriebsfähig ist, um die jeweiligen Dichtewerte jener Datenelemente ($D_{m,n}$, D_x) zu vergleichen und um in Abhängigkeit von dem Resultat solch eines Vergleichs solch ein Ausgabedatenelement ($O_{m,n}$) zu bestimmen, das dem betreffenden Objektpixel entspricht; und

ein Fehlerkorrekturmittel (14, 15), das betriebsfähig ist, um einen Objektpixelfehler ($E_{m,n}$) zu bestimmen, der für die Differenz von Dichtewerten zwischen dem genannten Bilddatenelement ($D_{m,n}$), das sich auf das genannte Objektpixel bezieht, und dem bestimmten Ausgabedatenelement ($O_{m,n}$) repräsentativ ist, und auch betriebsfähig ist, um in Abhängigkeit von jenem Fehler ($E_{m,n}$) solche weiteren Datenelemente in dem genannten Speichermittel (11) abzuwandeln, so daß die Schaltungsanordnung solch einen bestimmten Objektpixelfehler kompensieren kann, wenn ein nachfolgender Satz solcher Bilddatenelemente verarbeitet wird.

3. Bilddatenverarbeitungsschaltungsanordnung nach Anspruch 1 oder 2, bei der das genannte Fehlerkorrekturmittel (14, 15) eine Fehlerverteileinheit (15) enthält, die zum Empfangen des bestimmten Objektpixelfehlers ($E_{m,n}$) verbunden ist und auch zum Empfangen einer vorbestimmten Fehlerdiffusionsmatrix (K) verbunden ist, und betriebsfähig ist, um für jedes periphere Pixel des genannten Satzes, das dem genannten Objektpixel folgt, einen gewichteten Fehlerwert (zum Beispiel $K_{m-1,n-1} \times E_{m,n}$) zu berechnen, auf der Basis des bestimmten Objektpixelfehlers ($E_{m,n}$) und Wichtungswertes (zum Beispiel $K_{m-1,n-1}$) in der genannten Fehlerdiffusionsmatrix (K), der jenem peripheren Pixel entspricht;

welche Schaltungsanordnung ferner ein Datenelementabwandlungsmittel (17) enthält, das mit dem genannten Fehlerkorrekturmittel (14, 15) verbunden ist, zum Empfangen von diesem der jeweiligen gewichteten Fehlerwerte für die peripheren Pixels, die dem Objektpixel folgen, und betriebsfähig ist, um die weiteren Datenelemente, die sich auf jene Pixels beziehen, jeweilig in Abhängigkeit von den gewichteten Fehlerwerten abzuwandeln.

4. Bilddatenverarbeitungsschaltungsanordnung zum Verarbeiten von Bilddatenelementen, die für die jeweiligen Dichtewerte von Pixels, die ein Originalbild bilden, repräsentativ sind, um Ausgabedatenelemente zum Bestimmen der Dichtewerte von Pixels abzuleiten, die ein entsprechendes binäres Ausgabebild bilden, welche Schaltungsanordnung enthält:

ein Speichermittel (11) zum Speichern von sukzessiven Sätzen solcher Bilddatenelemente, die sich jeweilig auf solche Originalbildpixels beziehen, von welchen Sätzen jeder gebildet ist aus einem Datenelement ($D_{m,n}$), das sich auf ein Objektpixel bezieht, und weiteren Datenelementen ($D_{m-1,n-1}$, ... $D_{m+1,n+1}$), die sich jeweilig auf periphere Pixels beziehen, die an das genannte Objektpixel angrenzen;

ein Berechnungsmittel (12, 22), das mit dem genannten Speichermittel (11) verbunden ist und betriebsfähig ist, um Bilddatenelemente von solch einem gespeicherten Satz zu empfangen und um ein abgewandeltes Objektpixeldatenelement ($D_{m,n}$") zu erzeugen, das von dem genannten Bilddatenelement ($D_{m,n}$) abgeleitet ist, das sich auf das genannte Objektpixel des Satzes bezieht, in Abhängigkeit von den jeweiligen Differenzen von Dichtewerten zwischen jenem Objektpixelbilddatenelement ($D_{m,n}$) und solch weiteren Datenelementen ($D_{m-1,n}$, $D_{m,n-1}$, $D_{m,n+1}$, $D_{m+1,n}$) des betreffenden Satzes und auch in Abhängigkeit von einem Korrekturwert ($C_{m,n}$) für das Objektpixel des betreffenden Satzes;

ein Vergleichsmittel (13), das verbunden ist, um das abgewandelte Objektpixeldatenelement ($D_{m,n}$") zu empfangen, und betriebsfähig ist, um den Dichtewert von jenem Datenelement ($D_{m,n}$") mit einem voreingestellten Schwellenwert (TH_3) zu vergleichen und um in Abhängigkeit von dem Resultat solch eines Vergleichs solch ein Ausgabedatenelement ($O_{m,n}$) zu bestimmen, das dem betreffenden Objektpixel entspricht; und

ein Fehlerkorrekturmittel (14, 19-21), das betriebsfähig ist, um einen Objektpixelfehler ($E_{m,n}$) zu bestimmen, der für die Differenz von Dichtewerten zwischen dem genannten abgewandelten Objektpixeldatenelement ($D_{m,n}$") und dem bestimmten Ausgabedatenelement ($O_{m,n}$) repräsentativ ist, und auch betriebsfähig ist, um jenen Fehler zur Verwendung durch die Schaltungsanordnung beim Berechnen des Korrekturwertes für das Objektpixel eines nachfolgenden Satzes solcher Bilddatenelemente zu speichern, so daß die Schaltungsanordnung dadurch solch einen bestimmten Objektpixelfehler kompensieren kann, wenn jener nachfolgende Satz verarbeitet wird.

5. Bilddatenverarbeitungsschaltungsanordnung zum Verarbeiten von Bilddatenelementen, die für die jeweiligen Dich-

tewerte von Pixels, die ein Originalbild bilden, repräsentativ sind, um Ausgabedatenelemente zum Bestimmen der Dichtewerte von Pixels abzuleiten, die ein entsprechendes binäres Ausgabebild bilden, welche Schaltungsanordnung enthält:

ein Speichermittel (11) zum Speichern von sukzessiven Sätzen solcher Bilddatenelemente, die sich jeweilig auf solche Originalbildpixels beziehen, von welchen Sätzen jeder gebildet ist aus einem Datenelement ($D_{m,n}$), das sich auf ein Objektpixel bezieht, und weiteren Datenelementen ($D_{m-1,n-1}$, ..., $D_{m+1,n+1}$), die sich jeweilig auf periphere Pixels beziehen, die an das genannte Objektpixel angrenzen;
ein Berechnungsmittel (12, 22), das mit dem genannten Speichermittel (11) verbunden ist und betriebsfähig ist, um Bilddatenelemente solch eines gespeicherten Satzes zu empfangen und um ein Differenzdatenelement (D_x) in Abhängigkeit von den jeweiligen Differenzen von Dichtewerten zwischen dem Objektpixelbilddatenelement ($D_{m,n}$) des Satzes und solch weiteren Datenelementen des betreffenden Satzes zu erzeugen, und auch betriebsfähig ist, um ein korrigiertes Objektpixeldatenelement ($D_{m,n}'''$), das von dem genannten Objektpixeldatenelement ($D_{m,n}$) abgeleitet ist, in Abhängigkeit von einem Korrekturwert ($C_{m,n}$) für jenes Objektpixel zu erzeugen;
ein Vergleichsmittel (13), das verbunden ist, um das genannte korrigierte Objektpixeldatenelement ($D_{m,n}'''$) und das genannte Differenzdatenelement (D_x) zu empfangen, und betriebsfähig ist, um die jeweiligen Dichtewerte von jenen Datenelementen ($D_{m,n}'''$, D_x) zu vergleichen und um in Abhängigkeit von dem Resultat solch eines Vergleichs solch ein Ausgabedatenelement ($O_{m,n}$) zu bestimmen, das dem betreffenden Objektpixel entspricht; und
ein Fehlerkorrekturmittel (14, 15), das betriebsfähig ist, um einen Objektpixelfehler ($E_{m,n}$) zu bestimmen, der für die Differenz von Dichtewerten zwischen dem genannten korrigierten Objektpixeldatenelement ($D_{m,n}'''$) und dem bestimmten Ausgabedatenelement ($O_{m,n}$) repräsentativ ist, und auch betriebsfähig ist, um jenen Fehler zur Verwendung durch die Schaltungsanordnung beim Berechnen des Korrekturwertes für das Objektpixel eines nachfolgenden Satzes solcher Bilddatenelemente zu speichem, so daß die Schaltungsanordnung dadurch solch einen bestimmten Objektpixelfehler kompensieren kann, wenn jener nachfolgende Satz verarbeitet wird.

6. Bilddatenverarbeitungsschaltungsanordnung nach Anspruch 4 oder 5, bei der das genannte Fehlerkorrekturmittel (14, 19-21) betriebsfähig ist, um für jedes periphere Pixel des genannten Satzes, das dem genannten Objektpixel vorausgeht, den Objektpixelfehler ($E_{m-1,n-1}$, $E_{m-1,n}$, $E_{m-1,n+1}$, $E_{m,n-1}$) wiederaufzufinden, der durch das Fehlerkorrekturmittel beim Verarbeiten eines vorhergehenden Satzes der Bilddatenelemente, der als sein genanntes Objektpixel das betreffende periphere Pixel hatte, gespeichert wurde;

   welches Fehlerkorrekturmittel eine Durchschnittsfehlerberechnungseinheit enthält, die zum Empfangen einer vorbestimmten fehlergewichteten Matrix (K) verbunden ist und betriebsfähig ist, um die wiederaufgefundenen Fehler in Abhängigkeit von jeweiligen entsprechenden Wichtungswerten in der genannten fehlergewichteten Matrix (K) zu kombinieren, um den genannten Korrekturwert ($C_{m,n}$) zu berechnen.

Revendications

1. Circuit de traitement de données d'image pour le traitement d'éléments de données d'image représentant les valeurs respectives de densité des pixels constituant une image d'origine afin d'en déduire les éléments de données de sortie pour la détermination des valeurs de densité des pixels constituant une image binaire de sortie correspondante, circuit comprenant :

   - un moyen de stockage (11) pour le stockage d'ensembles successifs de tels éléments de données d'image correspondant respectivement à de tels pixels d'image d'origine, chacun des ensembles étant constitué d'un élément de données ($D_{m,n}$), concernant un pixel objet et des éléments de données supplémentaires ($D_{m-1,n-1}$, ..., $D_{m+1,n+1}$) concernant respectivement les pixels périphériques, adjacents audit pixel objet;
   - un moyen de calcul (12) raccordé audit moyen de stockage (11) et servant à recevoir les éléments de données d'image d'un tel ensemble stocké et à produire un élément modifié de données de pixel objet ($D_{m,n}'$) dérivé dudit élément de données d'image ($D_{m,n}$) concernant ledit pixel objet de l'ensemble selon les différences respectives des valeurs de densité entre cet élément de données d'image de pixel objet ($D_{m,n}$) et de tels éléments de données supplémentaires ($D_{m-1,n}$, $D_{m,n-1}$, $D_{m,n+1}$, $D_{m+1,n}$) de l'ensemble concerné;
   - un moyen de comparaison (13) raccordé pour recevoir l'élément modifié de données de pixel objet ($D_{m,n}'$) et servant à comparer la valeur de densité de cet élément de données ($D_{m,n}'$) avec une valeur de seuil préétablie

(TH_1) et selon le résultat d'une telle comparaison, servant à déterminer un tel élément de données de sortie ($O_{m,n}$) correspondant au pixel objet concerné; et

- des moyens de correction d'erreur (14, 15) servant à déterminer une erreur de pixel objet ($E_{m,n}$) représentant la différence des valeurs de densité entre ledit élément modifié de données de pixel objet ($D_{m,n}'$) et l'élément déterminé de données de sortie ($O_{m,n}'$) et servant de même, selon cette erreur ($E_{m,n}$), à modifier de tels éléments de données supplémentaires dans ledit moyen de stockage (11) de façon à permettre au circuit de compenser une telle erreur déterminée de pixel objet lors du traitement d'un ensemble suivant de tels éléments de données d'image.

2. Circuit de traitement de données d'image pour le traitement d'éléments de données d'image représentant les valeurs respectives de densité des pixels constituant une image d'origine afin d'en déduire les éléments de données de sortie pour la détermination des valeurs de densité des pixels constituant une image binaire de sortie correspondante, circuit comprenant :

- un moyen de stockage (11) pour le stockage d'ensembles successifs de tels éléments de données d'image correspondant respectivement à de tels pixels d'image d'origine, chacun des ensembles étant constitué d'un élément de données ($D_{m,n}$), concernant un pixel objet et des éléments de données supplémentaires ($D_{m-1,n-1}$, ..., $D_{m+1,n+1}$) concernant respectivement les pixels périphériques, adjacents audit pixel objet;
- un moyen de calcul (12) raccordé audit moyen de stockage (11) et prévu pour recevoir des éléments de données d'image d'un tel ensemble stocké et pour produire un élément de données de différence (D_x) selon les différences respectives des valeurs de densité entre l'élément de données de pixel objet ($D_{m,n}$) de l'ensemble et de tels éléments supplémentaires de données ($D_{m-1,n}$, $D_{m,n-1}$, $D_{m,n+1}$, $D_{m+1,n}$) de l'ensemble concerné;
- un moyen de comparaison (13) raccordé pour recevoir ledit élément de données d'image ($D_{m,n}$) concernant ledit pixel objet et ledit élément de données de différence (D_x) et servant à comparer les valeurs respectives de densité de ces éléments de données ($D_{m,n}$, D_x) et selon le résultat d'une telle comparaison, à déterminer un tel élément de données de sortie ($O_{m,n}$) correspondant au pixel objet concerné; et
- des moyens de correction d'erreur (14, 15) servant à déterminer une erreur de pixel objet ($E_{m,n}$) représentant la différence des valeurs de densité entre ledit élément de données d'image ($D_{m,n}$) concernant ledit pixel objet et l'élément déterminé de données de sortie ($O_{m,n}$) et servant, de même, selon cette erreur ($E_{m,n}$), à modifier de tels éléments de données supplémentaires dans ledit moyen de stockage (11) de façon à permettre au circuit de compenser une telle erreur déterminée de pixel objet lors du traitement d'un ensemble suivant de tels éléments de données d'image.

3. Circuit de traitement de données d'image selon la revendication 1 ou 2, dans lequel lesdits moyens de correction d'erreur (14, 15) comprennent une unité de répartition d'erreur (15) raccordée pour recevoir l'erreur déterminée de pixel objet ($E_{m,n}$) et raccordée, de même, pour recevoir une matrice de diffusion d'erreur prédéterminée (K) et servant, pour chaque pixel périphérique dudit ensemble suivant ledit pixel objet, à calculer une valeur d'erreur pondérée (par exemple, $K_{m-1,n-1}$ x $E_{m,n}$) sur la base de l'erreur déterminée de pixel objet ($E_{m,n}$) et de la valeur pondérée (par exemple, $K_{m-1,n-1}$) dans ladite matrice de diffusion d'erreur (K) correspondant à ce pixel périphérique;

circuit comprenant, de plus, un moyen de modification d'élément de données (17) raccordé auxdits moyens de correction d'erreur (14, 15) pour en recevoir les valeurs respectives d'erreur pondérées pour les pixels périphériques suivant le pixel objet et servant à modifier les éléments de données supplémentaires concernant ces pixels selon respectivement les valeurs d'erreur pondérées.

4. Circuit de traitement de données d'image pour le traitement d'éléments de données d'image représentant les valeurs respectives de densité des pixels constituant une image d'origine afin d'en déduire les éléments de données de sortie pour la détermination des valeurs de densité des pixels constituant une image binaire de sortie correspondante, circuit comprenant :

- un moyen de stockage (11) pour le stockage d'ensembles successifs de tels éléments de données d'image correspondant respectivement à de tels pixels d'image d'origine, chacun des ensembles étant constitué d'un élément de données ($D_{m,n}$), concernant un pixel objet et des éléments de données supplémentaires ($D_{m-1,n-1}$ ..., $D_{m+1,n+1}$) concernant respectivement les pixels périphériques, adjacents audit pixel objet;
- un moyen de calcul (12, 22) raccordé audit moyen de stockage (11) et servant à recevoir les éléments de données d'image d'un tel ensemble stocké et à produire un élément modifié de données de pixel objet ($D_{m,n}'$) dérivé dudit élément de données d'image ($D_{m,n}$) concernant ledit pixel objet de l'ensemble selon les différences respectives des valeurs de densité entre cet élément de données d'image de pixel objet ($D_{m,n}$) et de tels élé-

ments de données supplémentaires ($D_{m-1,n}$, $D_{m,n-1}$, $D_{m,n+1}$, $D_{m+1,n}$) de l'ensemble concerné et de même, selon une valeur de correction ($C_{m,n}$) pour le pixel objet de l'ensemble concerné;

- un moyen de comparaison (13) raccordé pour recevoir l'élément modifié de données de pixel objet ($D_{m,n}$") et servant à comparer la valeur de densité de cet élément de données ($D_{m,n}$") avec une valeur de seuil préétablie (TH_3) et selon le résultat d'une telle comparaison, servant à déterminer un tel élément de données de sortie ($O_{m,n}$) correspondant au pixel objet concerné; et
- des moyens de correction d'erreur (14, 19 à 21) servant à déterminer une erreur de pixel objet ($E_{m,n}$) représentant la différence des valeurs de densité entre ledit élément modifié de données de pixel objet ($D_{m,n}$") et l'élément déterminé de données de sortie ($O_{m,n}$) et servant, de même, à stocker cette erreur pour une utilisation par le circuit lors du calcul de la valeur de correction pour le pixel objet d'un ensemble suivant de tels éléments de données d'image, permettant ainsi au circuit de compenser une telle erreur déterminée de pixel objet lors du traitement de cet ensemble suivant.

5. Circuit de traitement de données d'image pour le traitement d'éléments de données d'image représentant les valeurs respectives de densité des pixels constituant une image d'origine afin d'en déduire les éléments de données de sortie pour la détermination des valeurs de densité des pixels constituant une image binaire de sortie correspondante, circuit comprenant :

- un moyen de stockage (11) pour le stockage d'ensembles successifs de tels éléments de données d'image correspondant respectivement à de tels pixels d'image d'origine, chacun des ensembles étant constitué d'un élément de données ($D_{m,n}$), concernant un pixel objet et des éléments de données supplémentaires ($D_{m-1,n-1}$, ..., $D_{m+1,n+1}$) concernant respectivement les pixels périphériques, adjacents audit pixel objet;
- des moyens de calcul (12, 22) raccordés audit moyen de stockage (11) et prévus pour recevoir des éléments de données d'image d'un tel ensemble stocké et pour produire un élément de données de différence (D_x) selon les différences respectives des valeurs de densité entre l'élément de données de pixel objet ($D_{m,n}$) de l'ensemble et de tels éléments supplémentaires de données de l'ensemble concerné et prévus, de même, pour produire un élément corrigé de données de pixel objet ($D_{m,n}$''') dérivé dudit élément de données de pixel objet ($D_{m,n}$) selon une valeur de correction ($C_{m,n}$) pour ce pixel objet;
- un moyen de comparaison (13) raccordé pour recevoir ledit élément corrigé de données de pixel objet ($D_{m,n}$''') et ledit élément de données de différence (D_x) et servant à comparer les valeurs respectives de densité de ces éléments de données ($D_{m,n}$''', D_x) et selon le résultat d'une telle comparaison, à déterminer un tel élément de données de sortie ($O_{m,n}$) correspondant au pixel objet concerné; et
- des moyens de correction d'erreur (14, 15) servant à déterminer une erreur de pixel objet ($E_{m,n}$) représentant la différence des valeurs de densité entre ledit élément corrigé de données de pixel objet ($D_{m,n}$''') et l'élément déterminé de données de sortie ($O_{m,n}$) et servant, de même, à stocker cette erreur pour une utilisation par le circuit lors du calcul de la valeur de correction pour le pixel objet d'un ensemble suivant de tels éléments de données d'image, permettant ainsi au circuit de compenser une telle erreur déterminée de pixel objet lors du traitement de cet ensemble suivant.

6. Circuit de traitement de données d'image selon la revendication 4 ou 5, dans lequel lesdits moyens de correction d'erreur (14, 19 à 21) sont prévus pour chaque pixel périphérique dudit ensemble précédant ledit pixel objet, pour extraire l'erreur de pixel objet ($E_{m-1,n-1}$, $E_{m-1,n}$, $E_{m-1,n+1}$, $E_{m,n-1}$) stockée par les moyens de correction d'erreur lors du traitement d'un ensemble précédent des éléments de données d'image présentant, comme son pixel objet, le pixel périphérique concerné;
lesdits moyens de correction d'erreur comprenant une unité de calcul d'erreur moyenne raccordée pour recevoir une matrice prédéterminée pondérée d'erreur (K) et servant à combiner les erreurs extraites selon les valeurs respectives correspondantes de pondération dans ladite matrice pondérée d'erreur (K) pour calculer ladite valeur de correction ($C_{m,n}$).

Fig. 1

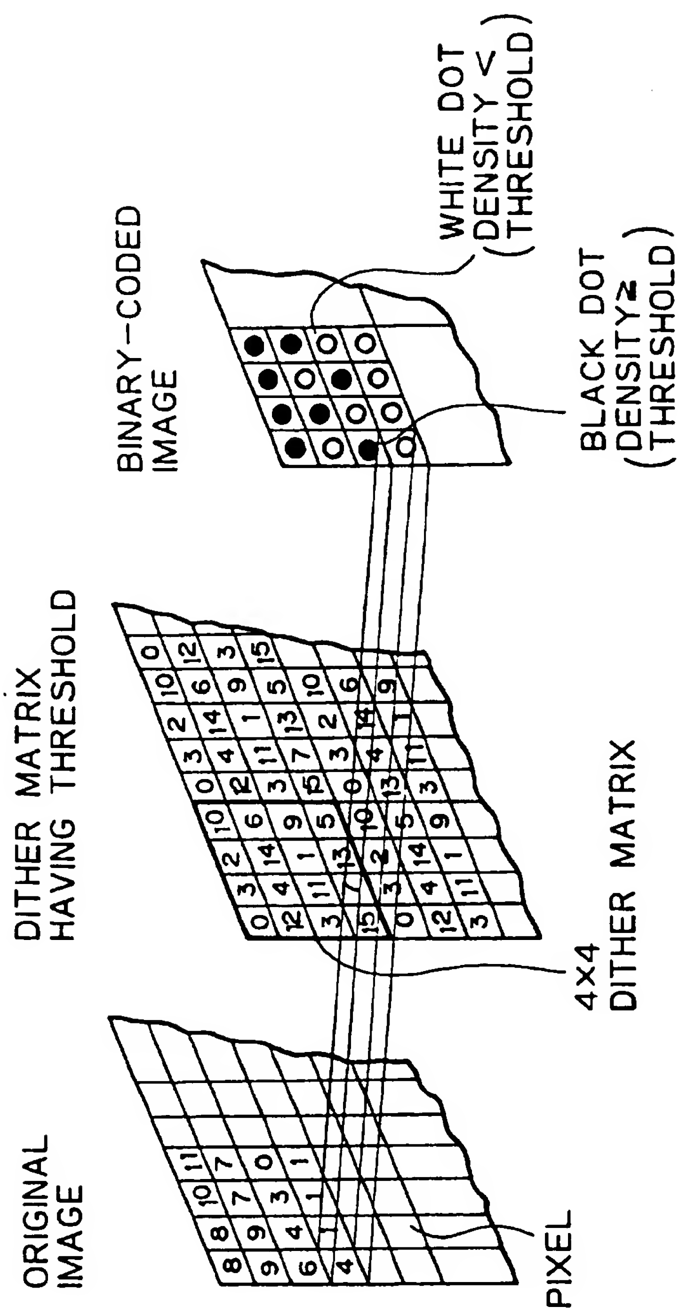

ORIGINAL IMAGE
PIXEL
DITHER MATRIX HAVING THRESHOLD
4x4 DITHER MATRIX
BINARY-CODED IMAGE
WHITE DOT (DENSITY < THRESHOLD)
BLACK DOT (DENSITY ≥ THRESHOLD)

Fig. 2A

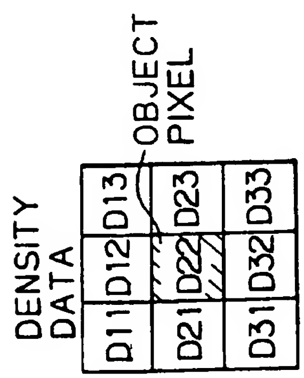


Fig. 2B

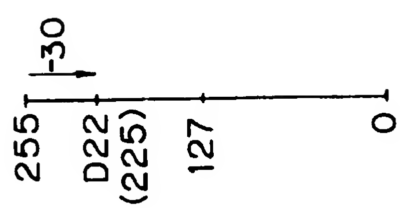


Fig. 2C

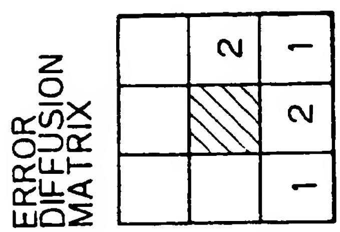


Fig. 3

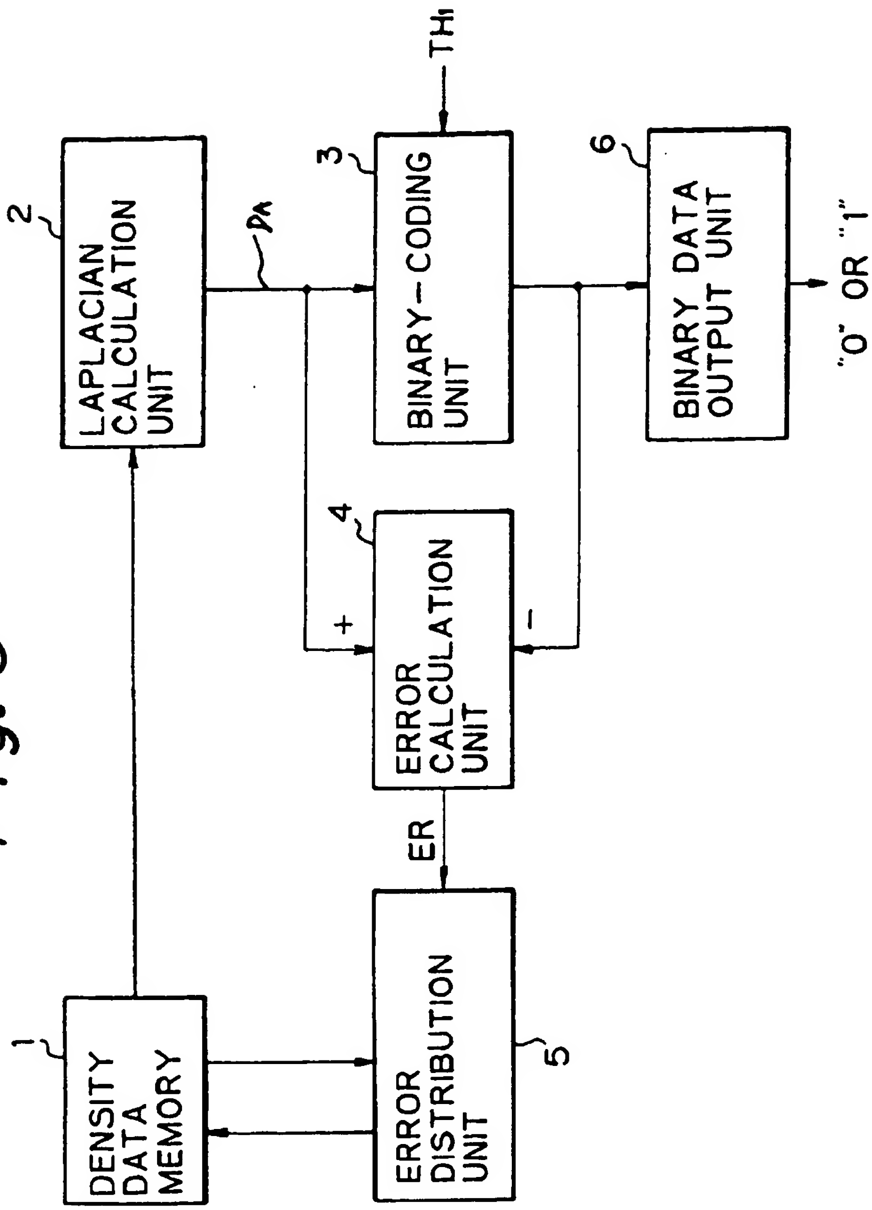

DENSITY DATA MEMORY
LAPLACIAN CALCULATION UNIT
BINARY-CODING UNIT
BINARY DATA OUTPUT UNIT
ERROR CALCULATION UNIT
ERROR DISTRIBUTION UNIT
TH1
"0" OR "1"
ER
+
−
1
2
3
4
5
6

Fig. 4

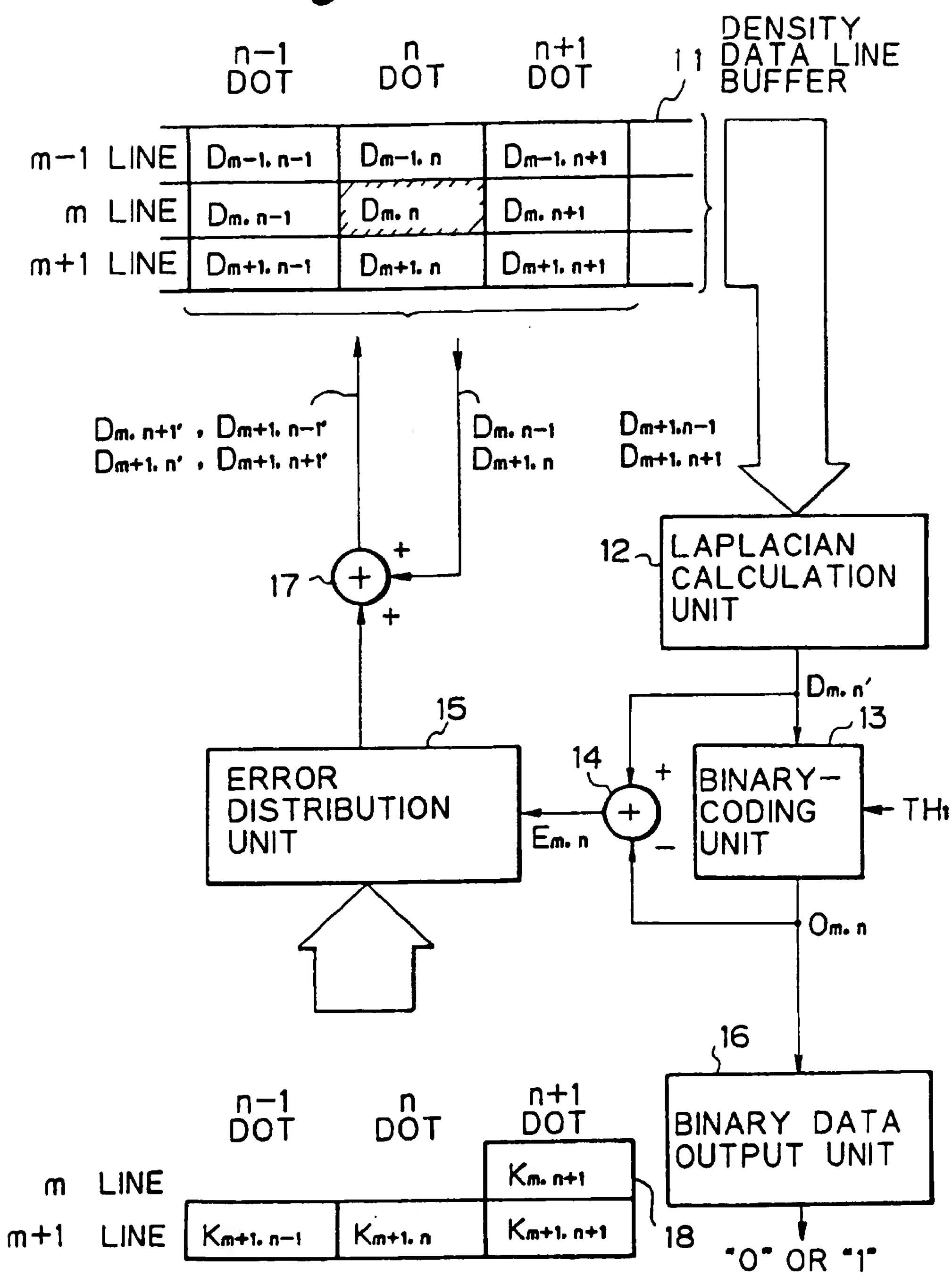

n−1 DOT
n DOT
n+1 DOT
11
DENSITY DATA LINE BUFFER
m−1 LINE
m LINE
m+1 LINE
Dm−1, n−1
Dm−1, n
Dm−1, n+1
Dm, n−1
Dm, n
Dm, n+1
Dm+1, n−1
Dm+1, n
Dm+1, n+1
Dm, n+1' , Dm+1, n−1'
Dm+1, n' , Dm+1, n+1'
Dm, n−1
Dm+1, n
Dm+1, n−1
Dm+1, n+1
17
12
LAPLACIAN CALCULATION UNIT
Dm, n'
13
15
14
ERROR DISTRIBUTION UNIT
Em, n
BINARY−CODING UNIT
TH1
Om, n
16
BINARY DATA OUTPUT UNIT
n−1 DOT
n DOT
n+1 DOT
m LINE
m+1 LINE
Km, n+1
Km+1, n−1
Km+1, n
Km+1, n+1
18
"0" OR "1"

Fig. 5

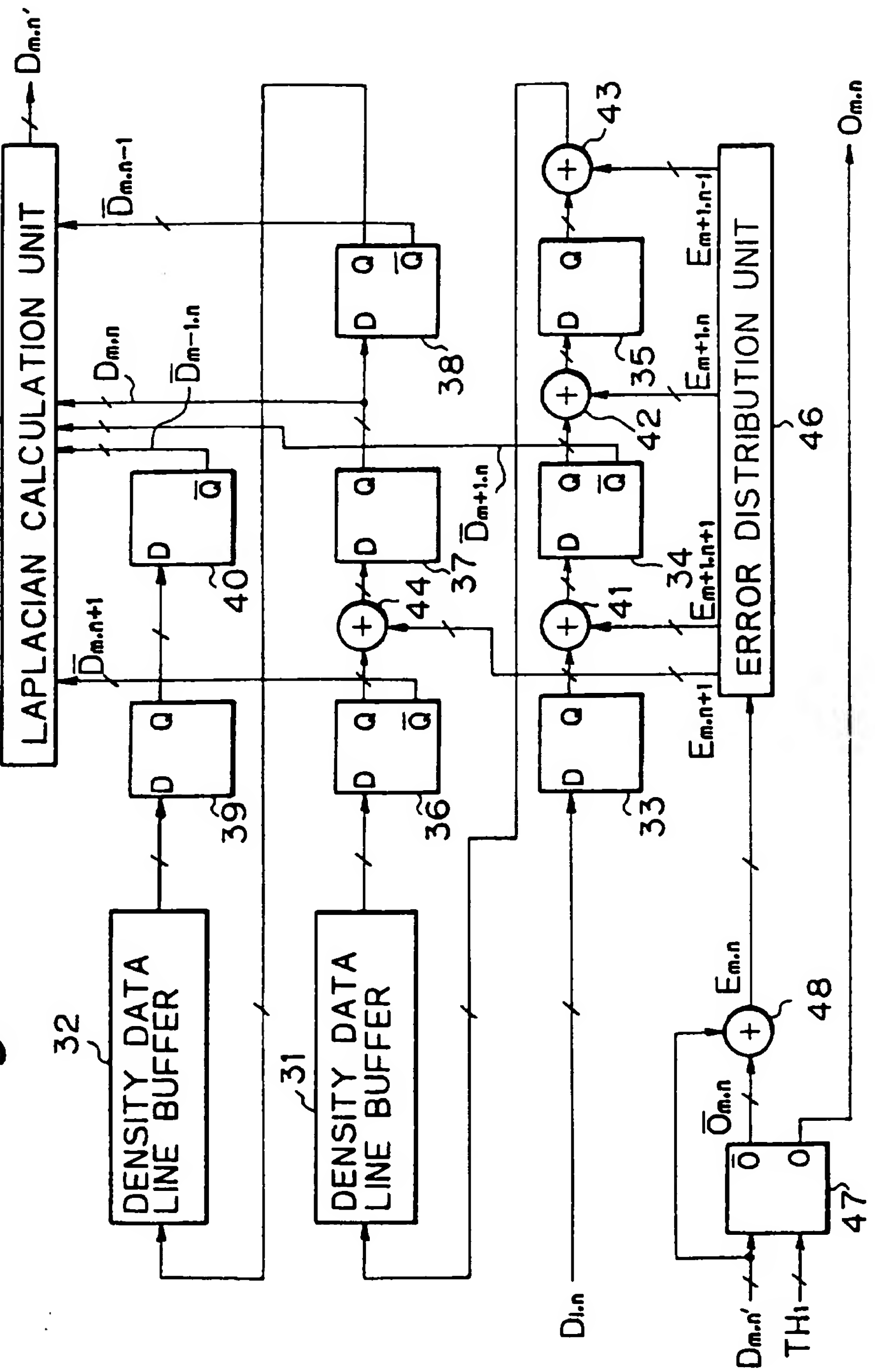

LAPLACIAN CALCULATION UNIT
DENSITY DATA LINE BUFFER
DENSITY DATA LINE BUFFER
ERROR DISTRIBUTION UNIT

Fig. 6

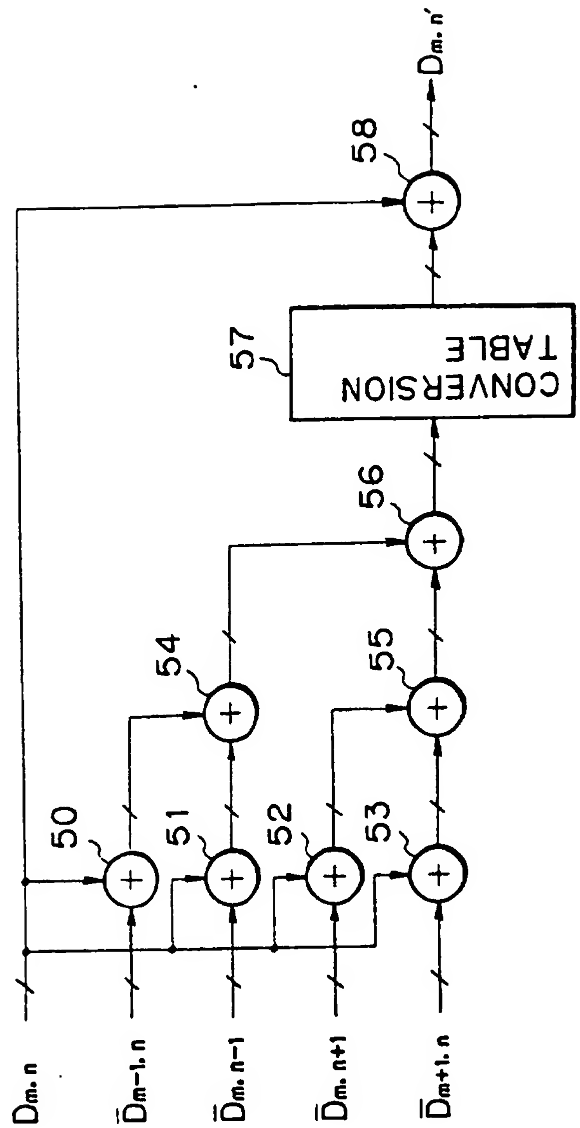


Fig. 7

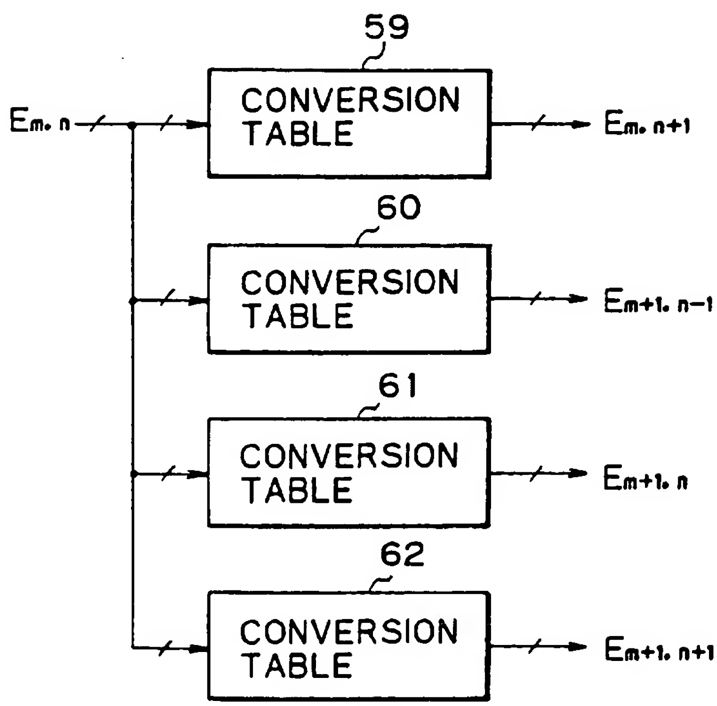

59
CONVERSION TABLE
60
CONVERSION TABLE
61
CONVERSION TABLE
62
CONVERSION TABLE
Em, n
Em, n+1
Em+1, n-1
Em+1, n
Em+1, n+1

Fig. 8A

0	−1	0
−1	+4	−1
0	−1	0

OBJECT PIXEL

COEFFICIENT

Fig. 8B

−1	0	−1
0	+4	0
−1	0	−1

Fig. 8C

0	0	−1	0	0
0	−3	−5	−3	0
−1	−5	+36	−5	−1
0	−3	−5	−3	0
0	0	−1	0	0

Fig. 9

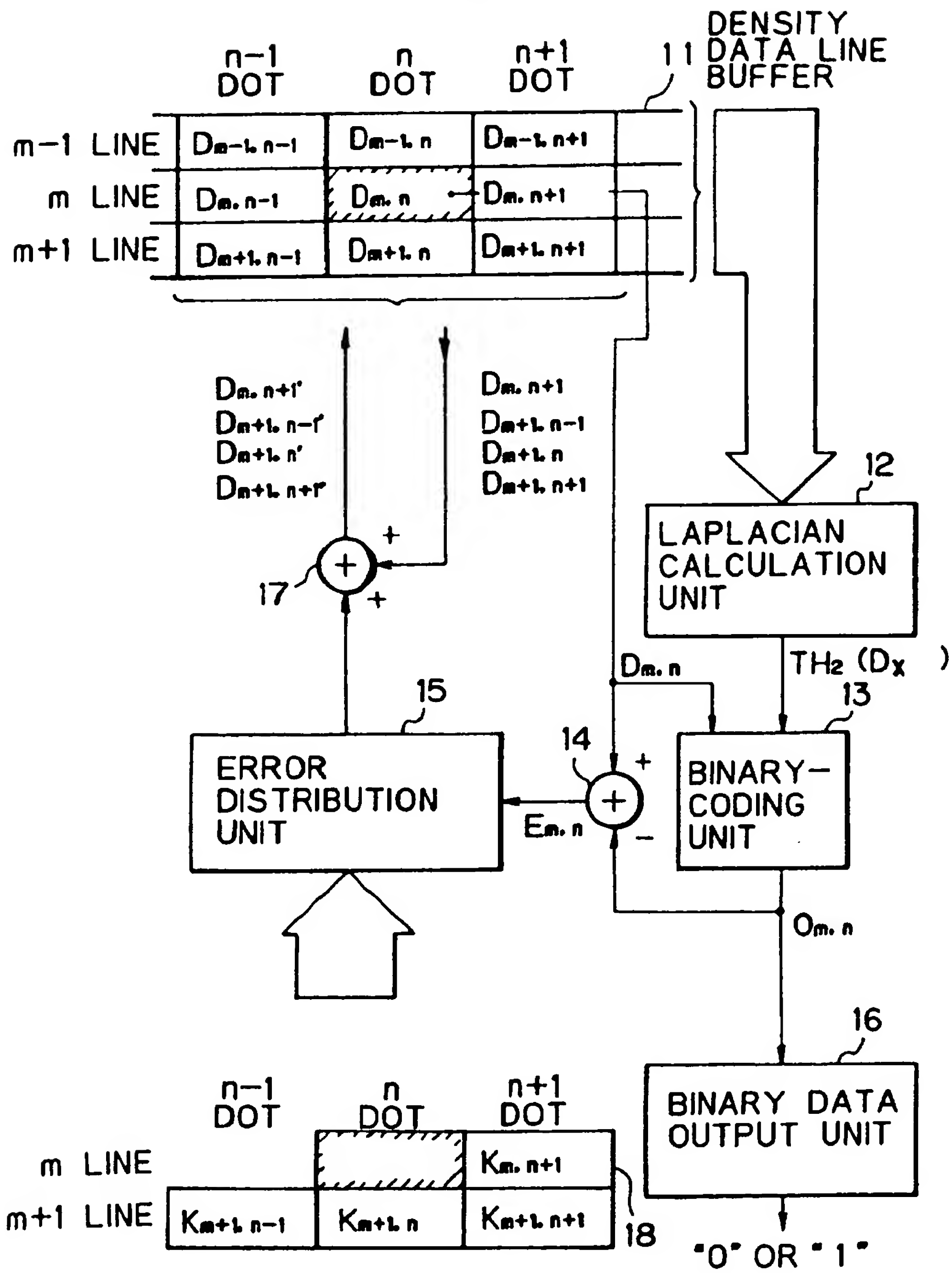

DENSITY DATA LINE BUFFER
11
n−1 DOT
n DOT
n+1 DOT
m−1 LINE
m LINE
m+1 LINE
Dm−1, n−1
Dm−1, n
Dm−1, n+1
Dm, n−1
Dm, n
Dm, n+1
Dm+1, n−1
Dm+1, n
Dm+1, n+1
Dm, n+1'
Dm+1, n−1'
Dm+1, n'
Dm+1, n+1'
Dm, n+1
Dm+1, n−1
Dm+1, n
Dm+1, n+1
12
LAPLACIAN CALCULATION UNIT
17
+
+
Dm, n
TH2 (Dx)
13
15
14
ERROR DISTRIBUTION UNIT
BINARY−CODING UNIT
Em, n
+
−
Om, n
16
BINARY DATA OUTPUT UNIT
"0" OR "1"
n−1 DOT
n DOT
n+1 DOT
m LINE
m+1 LINE
Km, n+1
Km+1, n−1
Km+1, n
Km+1, n+1
18

Fig. 10

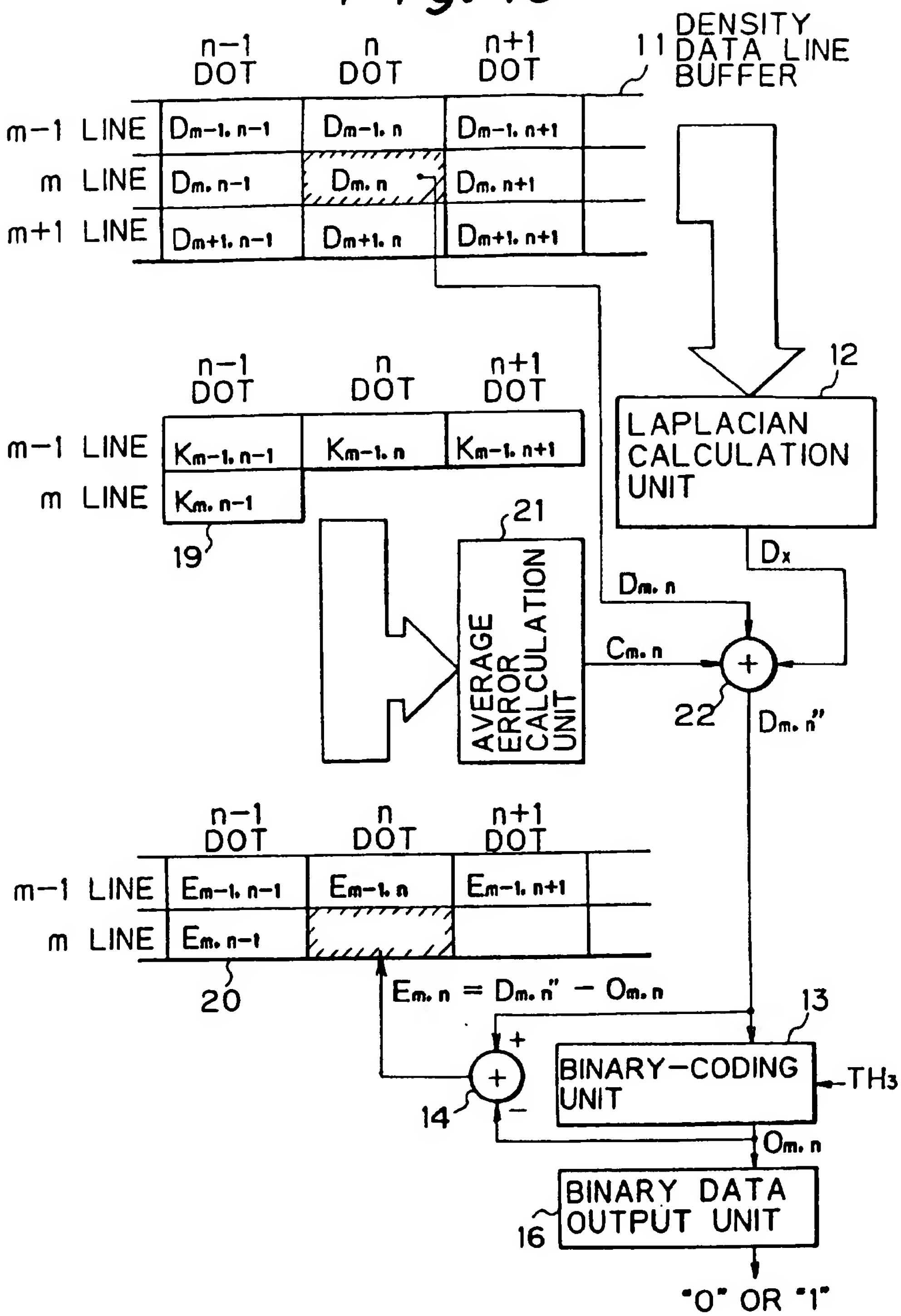
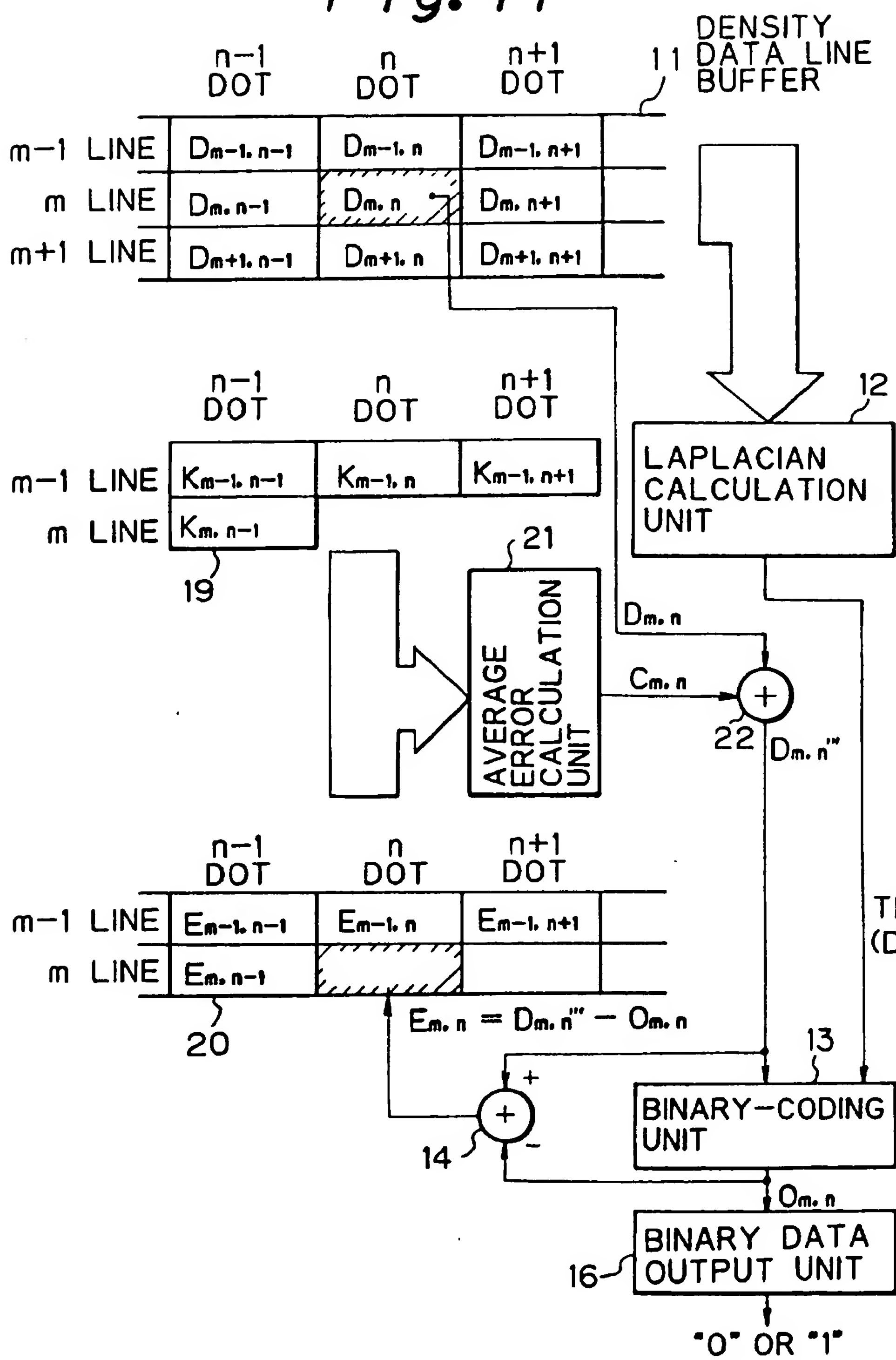

n−1 DOT
n DOT
n+1 DOT
11 DENSITY DATA LINE BUFFER
m−1 LINE
m LINE
m+1 LINE
Dm−1, n−1
Dm−1, n
Dm−1, n+1
Dm, n−1
Dm, n
Dm, n+1
Dm+1, n−1
Dm+1, n
Dm+1, n+1
12
LAPLACIAN CALCULATION UNIT
Km−1, n−1
Km−1, n
Km−1, n+1
Km, n−1
19
21
AVERAGE ERROR CALCULATION UNIT
Dx
Dm, n
Cm, n
22
Dm, n″
Em−1, n−1
Em−1, n
Em−1, n+1
Em, n−1
20
Em, n = Dm, n″ − Om, n
13
BINARY−CODING UNIT
TH3
14
Om, n
BINARY DATA OUTPUT UNIT
16
"0" OR "1"

Fig. 11


n−1 DOT
n DOT
n+1 DOT
11
DENSITY DATA LINE BUFFER
m−1 LINE
m LINE
m+1 LINE
Dm−1, n−1
Dm−1, n
Dm−1, n+1
Dm, n−1
Dm, n
Dm, n+1
Dm+1, n−1
Dm+1, n
Dm+1, n+1
12
LAPLACIAN CALCULATION UNIT
Km−1, n−1
Km−1, n
Km−1, n+1
Km, n−1
19
21
AVERAGE ERROR CALCULATION UNIT
Dm, n
Cm, n
22
Dm, n'''
Em−1, n−1
Em−1, n
Em−1, n+1
Em, n−1
20
Em, n = Dm, n''' − Om, n
TH4 (Dx)
13
BINARY−CODING UNIT
14
Om, n
BINARY DATA OUTPUT UNIT
16
"0" OR "1"